ISSN No. 0976-5697

Volume 1, No. 4, Nov-Dec 2010

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

Clustering Techniques in Web Content Mining

Mr.Ranjit R.Keole*
M E V[th] sem (CSE), Department of Computer Science & Engg., Prof. Ram Meghe Institute of Tech and Research, Badnera, Amaravati.(MS)
ranjitkeole@gmail.com

Dr.G.R.Bamnote
Prof. & Head of Department of Computer Science & Engg. Prof. Ram Meghe Institute of Tech and Research Badnera, Amaravati.(MS)
grbamnote@rediffmail.com

Abstract: Clustering is useful technique in the field of textual data mining. Cluster analysis divides objects into meaningful groups based on similarity between objects. Copious material is available from the World Wide Web (WWW) in response to any user-provided query. It becomes tedious for the user to manually extract real required information from this material. Large document collections, such as those delivered by Internet search engines, are difficult and time-consuming for users to read and analyze. The detection of common and distinctive topics within a document set, together with the generation of multi-document summaries, can greatly ease the burden of information management. This paper focus on this problem of mining the useful information from the collected web documents using fuzzy clustering of the text collected from the downloaded web documents.

Keywords- k-means, cure, birch, rock, erock, fuzzy clustering, text mining.

I. INTRODUCTION

With more than two billion pages created by millions of Web page authors and organizations, the World Wide Web is a tremendously rich knowledge base. The knowledge comes not only from the content of the pages themselves, but also from the unique characteristics of the Web, such as its hyperlink structure and its diversity of content and languages. A considerably large portion of information present on the World Wide Web (WWW) today is in the form of unstructured or semi-structured text data bases.

The WWW instantaneously delivers huge number of these documents in response to a user query. However, due to lack of structure, the users are at a loss to manage the information contained in these documents efficiently. The WWW continues to grow at an amazing rate as an informa tion gateway and as a medium for conducting business. Web mining is the extraction of interesting and useful knowledge and implicit information from atrifacts or activity related to the WWW.

In this context, the importance of data/text mining and knowledge discovery is increasing in different areas like: telecommunication, credit card services, sales and marketing etc [1]. Text mining is used to gather meaningful informa tion from text and includes tasks like Text Categorization, Text Clustering, Text Analysis and Docu ment Summariza tion. Text Mining examines unstructured textual information in an attempt to discover structure and implicit meanings within the text.

One main problem in this area of research is regarding organization of document data. This can be achieved by developing nomenclature or topics to identify different documents. However, assigning topics to documents in a large collection manually can prove to be an arduous task. We propose a technique to automatically cluster these documents into the related topics. Clustering is the proven technique for document grouping and categorization based on the similarity between these documents [2]. Documents within one cluster have high similarity with each another, but low similarity with documents in other clusters.

II. RELATED WORK

Document clustering has been widely applied in the field of information retrieval for improving search and retrieval efficiency [3]. Furthermore, document clustering has also been applied as a tool for browsing large document collections [4] and as a post-retrieval tool for organizing Web search results into meaningful groups [5]. Document clustering is recently applied to dynamically discover content relationships in e-Learning material based on document metadata descriptions [6, 12]. Main focus is on the discovery and representation of unobvious or unfamiliar knowledge about a domain rather than on facilitating the access to specific information resources through a set of document clusters.

Various techniques for accurate clustering have been proposed [13], e.g. K-MEAN [7, 14], CURE [8], BIRCH [9], ROCK [10]. K-MEAN clustering algorithm is used to partition objects into clusters while minimizing sum of distance between objects and their nearest center. In statistics and machine learning, *k*-means clustering is a method of cluster analysis which aims to partition observati ons into *k* clusters in which each observation belongs to the cluster with the nearest mean.

CURE (Clustering Using Representation) represents clusters by using multiple well scattered points called representatives. A constant number 'c' of well scattered points can be chosen from '2c' scattered points for merging two clusters. CURE can detect clusters with non spherical shapes and works well with outliers. CURE achieves this by representing each cluster by a certain fixed number of points that are generated by selecting well scattered points from the cluster and then shrinking them toward the center of the cluster by a specified fraction. Having more than one representative point per cluster allows CURE to adjust well to the geometry of non-spherical shapes and the shrinking helps to dampen the effects of outliers. To handle large databases, CURE employs a combination of random sampling and partitioning. A random sample drawn from the data set is first partitioned and each partition is partially

clustered. The partial clusters are then clustered in a second pass to yield the desired clusters. BIRCH (Balance and Iterative Reducing and Clustering Hierarchies) is useful algorithm for data represented in vector space. It also works well with outliers like CURE [11]. BIRCH incrementally and dynamically clusters incoming multi-dimensional metric data points to try to produce the best quality clustering with the available resources (i. e., available memory and time constraints). BIRCH can typically find a good clustering with a single scan of the data, and improve the quality further with a few additional scans. BIRCH is also the first clustering algorithm proposed in the database area to handle "noise)' (data points that are not part of the underlying pattern) effectively.

However, the traditional clustering algorithms fail while dealing with categorical attributes. As they are based on distance measure so their merging processing is not accurate in case of categorical data. ROCK (Robust Clustering Algorithm for Categorical Attributes) gives better quality clusters involving categorical data as compared with other traditional algorithms. Below we first describe the original ROCK approach and then propose our own enhancements to ROCK which we call the Enhanced ROCK or EROCK approach.

In hard clustering, data is divided into distinct clusters, where each data element belongs to exactly one cluster. In fuzzy clustering, data elements can belong to more than one cluster, and associated with each element is a set of membership levels. These indicate the strength of the association between that data element and a particular cluster. Fuzzy clustering is a process of assigning these membership levels, and then using them to assign data elements to one or more clusters.

Some of the clustering algorithms are discussed in the following sections.

III. FUZZY CLUSTERING OF TEXT

Topics that characterize a given knowledge domain are somehow associated with each other. Those topics may also be related to topics of other domains. Hence, documents may contain information that is relevant to different domains to some degree. With fuzzy clustering methods documents are attributed to several clusters simultaneously and thus, useful relationships between domains may be uncovered, which would otherwise be neglected by hard clustering methods.

A. FCM and H-FCM

Recently the Fuzzy c-Means (FCM) algorithm is modified for clustering text documents based on the cosine similarity coefficient rather than on the Euclidean distance. The modified algorithm works with normalized *k*-dimensional data vectors that lie in hyper-sphere of unit radius and hence has been named Hyper-spherical Fuzzy c-Means (H-FCM). The H-FCM algorithm for document clustering has shown that it outperforms the original FCM algorithm as well as the hard k-Means algorithm.

The objective function the H-FCM minimizes is similar to the FCM one, the difference being the replacement of the squared norm by a dissimilarity function $D_{i\alpha}$:

$$J_m\ (U,V) = \sum_{i=1}^{N}\sum_{\alpha=1}^{c} u_{\alpha i}{}^{m} D_{i\alpha} = \sum_{i=1}^{N}\sum_{\alpha=1}^{c} u_{\alpha i}{}^{m}\ (1 - \sum_{j=1}^{k} x_{ij}.v_{\alpha j}\ ) \tag{1}$$

The cosine coefficient ranges in the unit interval and when data vectors are normalized to unit length it is equivalent to the inner product. The dissimilarity function $D_{i\alpha}$ in equation (1) consists of a simple transformation of the cosine similarity coefficient, *i.e.* $D_{i\alpha} = 1 - S_{i\alpha}$.

$$u_{\alpha j} = \left[\sum_{\beta=1}^{c}\left(\frac{D_{i\alpha}}{D_{i\beta}}\right)^{\frac{1}{(m-1)}}\right]^{-1} = \left[\sum_{\beta=1}^{c}\left(\frac{1-\sum_{j=1}^{k} x_{ij}.v_{\propto j}}{1-\sum_{j=1}^{k} x_{ij}.v_{\beta j}}\right)^{\frac{1}{(m-1)}}\right]^{-1} \tag{2}$$

$$v_\alpha = \sum_{i=1}^{N} u_{\propto i}\ {}^{m}X_i\ .\left[\sum_{j=1}^{K}\left(\sum_{i=1}^{N} u_{\propto i}\ {}^{m}X_{ij}\right)^{2}\right]^{-1/2} \tag{3}$$

The update expression for the membership of data element x_i in cluster α, denoted as $u_{\alpha i}$ and shown in equation (2), is also similar to the original FCM expression since the calculation of $D_{i\alpha}$ does not depend explicitly on $u_{\alpha i}$. However, a new update expression for the cluster centroid v_α, shown in equation (3), had to be developed. Like the original algorithm, H-FCM runs iteratively until a local minimum of the objective function is found or the maximum number of iterations is reached**.**

B. Finding the optimum number of clusters

The H-FCM algorithm requires the selection of the number of clusters *c*. However, in most clustering applications the optimum *c* is not known *a priori*. A typical approach to find the best *c* is to run the clustering algorithm for a range of *c* values and then apply validity measures to determine which *c* leads to the best partition of the data set. The validity of individual clusters is usually evaluated based on their compactness and density. In low-dimensional spaces it is acceptable to assume that valid clusters are compact, dense and well separated from each other. However, text documents are typically represented as high-dimensional sparse vectors. In such problem space, the similarity between documents and cluster centroids is generally low and hence, compact clusters are not expected. Therefore, the approach mentioned above for finding the optimum *c* is inappropriate. A question that arises is how the H-FCM algorithm is able to discover meaningful document clusters considering such low similarity patterns. As observed for the hard k-Means algorithm, the good performance of the H-FCM is justified by the fact that documents within a given cluster are always more similar to the corresponding centroid than documents outside that cluster, regardless of the number of clusters that has been

selected. It is believe that in the high-dimensional document space the issue of finding the optimum number of clusters is not so relevant. The choice of *c* should rather address the desired granularity level, since the higher the number of clusters the more specific will be the topics covered by the documents in those clusters.

IV. ROCK ALGORITHM

The original algorithm used the Jaccard coefficient for similarity measure but later on a new technique was introduced according to which two points are considered similar if they share a large enough number of neighbors. The basic steps of ROCK algorithm are:

A. Obtain a random sample of points from the data set 'S'.

B. Compute the link value for each pair of points using the

$$Sim\ (\ T_1\ ,T_2\ ) = \frac{|T_1 \cap\ T_1|}{|T_1 \cup\ T_1|}$$

Jaccard coefficient:

C. Maintain a heap (as a sparse matrix) for each cluster's links.

D. Perform an agglomerative hierarchical clustering on data using number of shared objects (as indicated by the Jaccard coefficient) as clustering criterion.

E. Assign the remaining points to the found cluster.

F. Repeat steps 1-5 until the required number of clusters has been found.

ROCK algorithm has following advantages over other clustering algorithms:

A. It works well for the categorical data.

B. Once a document has been added to a specific cluster, it will not be re-assigned to another cluster at the same level of hierarchy. In other words, document switching across the clusters is avoided using ROCK.

C. It uses the concept of links instead of using distance formula for measuring similarity resulting in more flexible clustering.

D. It generates better quality clusters than other algorithms.

Limitations of ROCK include the following:

A. ROCK algorithm used sparse matrix for storing cluster links.

B. Sparse matrix takes more space so efficiency suffers adversely.

C. Similarity is calculated by using Jaccard coefficient.

D. Similarity function is dependent on document length.

a) Enhancement ROCK

EROCK approach includes several enhancements to overcome the limitations of the ROCK algorithm. Here we discuss these enhancements. First, ROCK algorithm draws random sample from the database. It then calculates links between the points in the sample. The purposed approach (EROCK) makes use of entire data base for clustering. Every point in the database is treated as a separate cluster meaning that every document is treated as a cluster. Then the links between these clusters are calculated. The clusters with the highest number of links are then merged. This process goes on until the specified numbers of clusters are formed. So by decomposing the whole database, linkage and topic generation will become efficient. Second, ROCK algorithm uses similarity measure based on Jaccard coefficient. The proposed cosine measure:

$$CosSim\ (\ V_1\ ,V_2\ ) = \frac{|V_1\ .\ V_2|}{|V_1|\ \ |V_2|}$$

Where v1 and v2 are the term frequency vectors. v1. v2 is the vector dot product defined as:

$$\sum_{i=1}^{k} i = V_1 i \quad V_2 i$$

And |v1| is defined as:

$$|V_1| = \sqrt{V_1 . V_2}$$

Cosine similarity is independent of the document length. Due to this property processing becomes efficient. Cosine similarity has advantages over Euclidean distance while applied on large documents (when documents tends of scale up), Euclidean will be preferred otherwise. Third, ROCK uses sparse matrix for link information. The sparse matrix requires more space and long list of references because of which efficiency suffers adversely. In EROCK adjacency list instead of sparse matrix is proposed for maintaining link information between neighboring clusters. Adjacency list is a preferred data structure when data is large and sparse. Adjacency list keeps track of only neighboring documents and utilizes lesser space as compared to sparse matrix. Besides space efficiency it is easier to find all vertices adjacent to a given vertex in a list.

Inputs: The EROCK algorithm requires some initial parameters which are necessary for the whole process. Following are the major inputs to run the algorithm:

A. A directory containing text documents (Corpus).

B. Threshold for number of clusters to be formed.

C. Threshold value for measuring similarity of documents.

D. Threshold value for taking top most frequent words for labeling the folders.

b) Document Clustering and Topic Generation Using EROCK Algorithm

Basic steps of EROCK are the same as those of ROCK. For document clustering and topic generation, the text files in the corpus are first converted into documents. Following are the steps involved in making the clusters, using EROCK algorithm:

(***a).*** Build documents from the text file present in the specified folder.

(b). Compute links of every document with every other document using cosine similarity measure.

(c). Maintain neighbors of each document in an adjacency list structure.

(d). After computing links for all documents, each document is treated as a cluster.

(e). Extract the best two clusters that will be merged to form one cluster. This decision is made on the basis of goodness measures. In EROCK, goodness measure defined as the two clusters which have maximum number of links between them. Let these two clusters be u and v.

(f). Now merge the two clusters u and v. Merging of two clusters involve, merging the names of the two clusters, the documents of two clusters and links of two clusters. This will result in a merged cluster called w.

(g). For each cluster x that belongs to the link of w take following steps:

i. Remove clusters *u* and *v* from the links of *x*.
ii. Calculate the link count for *w* with respect to *x*.
iii. Add cluster *w* to the link of *x*.
iv. Add cluster *x* to the link of *w*.
v. Update cluster *x* in the original cluster list.
vi. Add cluster *x* to the original cluster list
vii. Repeat step (iv.) until the required number of clusters are formed or there are no two clusters found to be merged.
viii. After obtaining the final merged cluster list apply labeling process on each. For labeling, the most frequent word from each document of a cluster is used. Take top most frequent words based on the threshold value.

The word with high frequency will be treated as the topic or label for a cluster. All related documents will be placed under one topic. Physically these documents will be put in folders with topics or label as folder name.

Output:

(a) A list of clusters labeled properly.

(b) Each cluster gets converted into a physical folder/directory on the disk and each folder contains the documents of the respective cluster.

V. CONCLUSION AND FUTURE WORK

In this paper, we surveyed K-MEANS, CURE, BIRCH, ROCK, EROCK and a novel fuzzy clustering algorithm H-FCM for text mining.

The H-FCM generates clusters with a higher level of granularity and that the resulting clusters hierarchy successfully links clusters of the same topic. Also it has been found that by enhancing some parameters of traditional ROCK algorithm, we can get better results. The experimental results obtained from the research are very encouraging. The outcome of these experiments shows that by using EROCK approach, the cumbersome task of manually grouping and arranging files becomes very easy. Now user will be able to get relevant information easily without doing tedious manual activity. Huge information is now available in the form of text documents so documents/clusters having related information are grouped together and labeled accordingly. Clusters are merged only if closeness and inter connectivity of items within both clusters are of high significance. Finally it is observed that EROCK gives good performance for large datasets.

There are many areas in text mining; where one may carry on work to enhance those areas. Out of these, the labeling of the clusters is a very daunting challenge of this time. No remarkable effort has been made in this regard to get good result. That is why automatic labeling of the clusters is not so much accurate. A keen and concerted work has been done to remove this hurdle. It will certainly serve as a lime length for future researchers.

VI. REFERENCES

[1] Hsinchun Chen and Michael Chau, "Web Mining: Machine learning for Web Applications", Annual Review of Information Science and Technology 2003.

[2] Mr. Rizwan Ahmad and Dr. Aasia Khanum, "Document Topic Generation in Text Mining by Using Cluster Analysis with EROCK", Internatio nal Journal of Computer Science & Security (IJCSS), Volume (4) : Issue (2) Aug 2008.

[3] M.E.S. Mendes Rodrigues and L. Sacks, "A Scalable Hierarchical Fuzzy Clustering Algorithm for Text Mining", Department of Electronic and Electrical Engineering University College London Torrington Place, London, WC1E 7JE, United Kingdom, 2004.

[4] D.R. Cutting, D.R. Karger, J.O. Pederson and J.W. Tukey (1992). Scatter/gather: a cluster-based approach to browsing large document collections. In: Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'92, pp. 318-329, Copenhagen, Denmark, June 1992.

[5] A. Schenker, M. Last and A. Kandel (2001). A term-based algorithm for hierarchical clustering of Web documents. In: Proceedings of the Joint 9th IFSA World Congress and 20th NAFIPS International Conference, vol.5, pp. 3076-3081, Vancouver, Canada, July 2001.

[6] M.E.S. Mendes, W. Jarrett, O. Prnjat and L. Sacks (2003). Flexible searching and browsing for telecoms learning material. In: Proceedings of the 2003 International Symposium on Telecommunica tions, IST'2003, Isfahan, Iran, August 2003.

[7] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, Angela Y. Wu, "An Efficient k-Means Clustering Algorithm: Analysis and Implementation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 7, July 2002.

[8] Linas Baltruns, Juozas Gordevicius, "Implementa tion of CURE Clustering Algorithm", SIGMOD Seattle, WA, USA ACM February 1, 2005.

[9] Tian Zhang, Raghu Ramakrishan, Miron Livny, "BIRCH: An Efficent Data Clustering Method for Very Large Databases" SIGMOD '96 6/96 Montreal, Canada IQ 1996 ACM.

[10] Shaoxu Song and Chunping Li, "Improved ROCK for Text Clustering Using Asymmetric Proximity", SOFSEM 2006, LNCS 3831, pp. 501–510, 2006.

[11] Linas Baltruns, Juozas Gordevicius, "Implementa tion of CURE Clustering Algorithm", February 1, 2005.

[12] Raymond Y.K. Lau, Senior Member, IEEE, Dawei Song, Yuefeng Li, Member, IEEE, Terence C.H. Cheung, Member, IEEE, and Jin-Xing Hao, "Toward a Fuzzy Domain Ontology Extraction Method for Adaptive e-Learning," IEEE Trans. On Knowledge and Data Engineering, Vol. 21, No. 6, Jun 2009.

[13] Shady Shehata, Member, IEEE, Fakhri Karray, Senior Member, IEEE, and Mohamed S. Kamel, Fellow, IEEE, "An Efficient Concept-Based Mining Model for Enhancing Text Clustering", IEEE Trans. On Knowledge and Data Engineering, Vol. 22, No. 10, Oct 2010.

[14] Jingwen Tian, Meijuan Gao, and Yang Sun, "Study on Web Classification Mining Method Based on Fuzzy Neural Network",Proceedings f the IEEE International Conference on Automation and Logistics Shenyang, China August 2009.

